A METHOD OF CONTROLLING THE EXECUTION OF A REQUEST FOR ACTIONS TRANSMITTED BY A SERVER TO A CHIP CARD VIA A TERMINAL

INSB1

The present invention relates to systems for exchanges of messages between an application server and chip cards using a communication network. It applies to exchanges taking place through telecommunication networks, switched telephone networks, cellular networks or the Internet.

Generally the messages exchanged between an application server and the corresponding application in a chip card pass through intermediate equipment which will be referred to as a terminal hereinafter. The chip card of a user co-operates with the terminal to allow the exchanges.

Where the network used is a telephony network, the terminal is a telecommunications terminal. Where the network used is a computer network, the terminal is

data processing equipment of the computer type equipped with an interface for reading/writing to chip cards.

A server under the control of a card-issuing organisation wishing to effect a protected action in a chip card (or in an application of the ~~said~~ card) via a telephone network uses cryptographic certificates for ensuring the security of the exchanges.

However, in the event of the loss of a message during the transmission or execution or in the case of an attempted fraud, the re-synchronisation of the server/card messages can pose security problems.

Where the terminal is a dedicated protected terminal under the control of the issuing body (for example an automatic note dispenser AND under the control of a bank), the loss of a message is compensated for by synchronisation messages using both the software of the server and the software of the dedicated terminal. The dedicated terminal is protected either physically (AND) or contains inside it an SAM (Secure Authentication Module), and in all cases is closely monitored by the issuing organisation.

If the terminal used is not a dedicated protected terminal (for example a GSM telephone, PC on the Internet, etc), the synchronisation mechanisms cannot be based on the security of the terminal, because the latter cannot be monitored by the issuer.

In fact it is important to be able to re-synchronise the source of the messages and the chip card in the event of any transmission problem on the

network. This problem has been posed in terms of security vis-à-vis operators and service providers.

At the present time there is no system designed to ensure synchronisation between the card and server, in cases where during a current transaction, consequently accepted by the card, the server profits from the connection to send a message containing one or more actions to be implemented by the card, these actions being able, for example, to be a re-charging of units of value or parameters (monetary or other) or a loading of a new application.

In fact, provision is made, in the more general context of multi-application cards, for messages to be sent when the user has made a transaction request in order to send commands for actions to be undertaken during the running of the application for the current transaction.

Such messages will for example make it possible to control an electric purse recharging in the case of an electronic purse application, or to modify banking parameters of the bank application, or the loading of a new application into the card.

It is clear that, in this situation, the server will not be informed where the ~~card~~ message is lost.

In other words, performing protected actions on a non-dedicated terminal is feasible today but requires either high user constraints (cards or applications blocked if the security action has not ended), or risks of loss of information (for example loss of a transaction for recharging an electronic purse).

a

Summary of the Invention

The purpose of the invention is that the server can detect failures in execution of one or more actions or commands, linked to a loss of messages between the server and chip card or to failure to execute actions in the card, the said messages having been transmitted to the card, possibly during a current transaction, in order to inform the server thereof so that the latter determines what are the last actions or commands not executed by the card.

According to a procedure pre-established in accordance with the action or actions not implemented, the server can for example send back the message containing the said action or actions and allow their execution.

To this end, the object of the invention is particularly a method of monitoring an execution of a request for actions transmitted by a server to a card via a terminal, the said card including an action counter, characterised in that it includes the following steps;

a) on the sending by the server of a message including a request comprising one or more actions to be implemented by the card, the server stores the number n of actions in the request;

b) on reception of the message, the card successively executes the action or actions in the request whilst incrementing its action counter between each action if the action is properly executed and refusing this action and the successive actions if the

action has not been correctly executed without incrementing its counter;

c) the variation between the value in the card and the one stored in the server are compared and it is determined that the last x actions (commands) are not executed if the result of the comparison has a difference of x.

The incrementation of the action counter corresponds to the number of actions correctly executed.

The number x is equal to 0 if all the actions are correctly executed; this number x can therefore vary from 1 to n if the last or all the actions have failed.

To compare the variation between the value in the card and the one stored in the server, the card transmits to the server the current value of its counter before and after execution of the action command.

To compare the variation between the value in the card and the one stored in the server, the card calculates the value of the variation of its counter following the execution of the action command and transmits it to the server.

According to another characteristic, any exchange of the value of the action counter of the card is effected systematically in a protected manner.

To this end, the last value of the action counter of the card is transmitted with a cryptogram whose calculation involves the said last value.

According to another characteristic the current last value of the action counter in the card is transmitted to the server in real time, that is to say during the current transaction.

According to one example the value can be transmitted by means of the message acknowledging the current transaction in the card.

According to another characteristic the value of the card action counter is transmitted to the server in non real time.

According to one example the value of the action counter can be transmitted by means of a message of a new request for a transaction by the card by the server.

According to another example the value of the card action counter is transmitted by means of an information method sent from the card to the server.

Another object of the invention is a card for implementing the aforementioned method including a counter and means of managing this counter, characterised in that the said management means are able to increment the said action counter between each action if the action is correctly executed and not to increment it for this action nor for the following actions if this action has not been executed.

Brief Description of the Drawings

Other characteristics and advantages of the present invention will emerge from the reading of the following description given below by way of non-limited example and with regard to the drawings in which:

- Figure 1 illustrates message exchanges between server and chip card according to the invention,

- Figure 2 illustrates in detail message exchanges between server and chip card in the case of the loss of a message,

- Figure 3 illustrates another case of the loss of a message.

Detailed Description

Action request means a message containing a set of n commands, n of course being able to be equal to 1.

For a better understanding of the remainder of the description, reference can be made to the diagram in Figure 1.

Throughout the remainder, the case where the server 2 profits from a current transaction in a card 1 in order to send it a request containing one or more actions which the card is to execute has been taken as an example.

Naturally in this case an action request will be sent with the response to the current transaction if the said transaction requires a response. If such is not the case, a response is created containing solely the action request. The terminal which is in communication with the server receives the message corresponding to this response, and removes the envelope from this message in order to transmit the actions to the card.

An action request can include several actions to be undertaken by the card, that is to say, as stated at the beginning of the description, a set of n commands.

By way of example an action request can be a request to change one or more parameters in an application program or the loading of a new application or the charging of units of value.

The change of a parameter corresponds to an action from the card which is an operation of erasure and writing at a predetermined address.

The change of several parameters corresponds to as many erasure and writing operations at distinct addresses as there are parameters and consequently to as many actions to be undertaken as there are parameters to be changed.

Details will now be given of what occurs on the card side and the server side.

Card side:

The card 1 increments, after each correctly performed action, the action counter CA as soon as it receives from the server one or more actions to be undertaken and as soon as it has been able to successfully execute each of these actions.

The value of the counter is sent back to the server, for example each time the card sends a message to the server (message 3 or message 4 in Figure 1).

The value of the counter can be sent back to the server 2 essentially at the time of the following actions:

- when there is a transaction acknowledgement (if during a transaction an acknowledgement message is sent back to the server the value of the action counter can be put in this acknowledgement) (example: message 3),

- when there is a transaction request or card authentication request to the server (example: message 4),

- in the case of bank cards or electronic purses,

- every past transaction is stored in each terminal 3,

- the stored transaction is sent back to the server so that the server can trigger the process of paying the merchant with whom the transaction took place, the action counter CA can be sent back with this transaction.

Thus the value of the content of the action counter is always sent back to the server either in real time when this is done at the time of acknowledgement or in non real time when there is a new transaction request or when a transaction storage is sent back.

Server side:

For each card containing an application which is dedicated to it having a current action request, the server must store:

- the identification number of the application,
- the current value of the action counter,
- the list of current actions for this card.

Thus the server to which there belongs an application placed in a multi-application chip card can, during any transaction requested by the card, demand an action such as a recharging of units, or a loading of a program or a loading of new parameters for a program resident in the card.

The server can thus send actions to the card by a script mechanism which cannot be interpreted by the terminal 3, which is situated between the server and the card in order to provide communication. The terminal 3 transmits the message or messages received in the script to the card in a transparent manner.

Details will now be given of all the processing in the case where the sending back of the content of the action counter takes place in real time and in the case where everything occurs correctly, that is to say in the case where there is no loss of message and where the execution by the card has taken place correctly.

Reference can be made to the particular embodiment illustrated by the diagram in Figure 1 to give a better understanding.

- At time dti the bearer requests, via his terminal 3, a transaction (a payment or another transaction): message 1.

- The card prepares the transaction and a cryptogram, that is to say the authentication data, designated hereafter as MAC, and transmits to the terminal.

Associated with this transaction, the banking application joins the current value of CA of its action counter protected by the cryptogram.

- The terminal sends back the transaction to the bank server.

In practical terms, the card sends a transaction request message containing the data MAC1 and the value

of the action counter CA, and the identification of the requested transaction.

The server verifies the authentication data for the card MAC1 and processes the transaction. The server can at this moment perform an action in the card application.

In a particular example, it may be a case of the loading of a monetary parameter into the card, but, as stated, other actions of the electronic purse recharging type are also possible.

- For this purpose, the server will prepare one or more parameter loading commands contained in an information field in an action referred to as script 1, and the security authentication data MAC2.

- The action request is sent by means of a message 2 which can contain the response to the current transaction if such a response is provided for the application concerned.

At the time of the sending of script 1 to the card, the server stores this script 1 in a database, associating therewith the data relating to the card, as well as the current value CA of the action counter of the card (sent from the card to the server during the transaction request). This information will make it possible to effect the server-card synchronisation.

- The card, which receives the commands one by one from script 1, verifies the cryptogram MAC2, and atomically (that is to say on a single occasion and indivisibly) performs action by action in the list of script 1 and increments the content CA of the counter

after each action if this has occurred correctly. When an action has occurred incorrectly, the action counter is not incremented and the other actions are not accepted.

- In order to send to the server the new value CA' of the action counter CA of the card, several schemes are possible:

  - sending back during a message acknowledging the current transaction, that is to say in real time (corresponds to the message 3 of the current transaction);

  - sending back the value of CA' during the next transaction (corresponds to the message 4 occurring at time dtj);

  - at any time, that is to say when the card sends information to the server.

- In the case of the example described, the card sends a protected acknowledgement to the server including the content CA' in real time. This can then compare the value returned by the acknowledgement with the value stored in its base.

If the value of CA' = CA+n, n being the number of actions of the script 1, this proves that the script 1 has been run correctly in the card. The server can then erase this script in the database.

A description will now be given, in relation to Figure 2, taking the same example, of what happens when a cutoff or a loss of action request message (message 2) occurs.

In this case, the command script 1 has not arrived in the card. The server will have to re-synchronise itself. The server is informed of this situation since according to this example it has not received an acknowledgement.

Where the server is not awaiting an acknowledgement, it is informed of when it receives the last of the card action counter, that is to say for example at the next transaction.

In fact, during the authentication of the card by the server (verification MAC1), the server identifies that this card has not received script 1 (or that script 1 has not been effected correctly in the card) by means of the value of CA' of the action counter which is sent back to the server and compared with the value of CA stored in the server.

If CA' is less than CA and not equal, this means that the last action or actions have not been performed correctly.

In this case the server updates its database DB, erasing the value of CA in order to put in the value of CA'. The server is once again synchronised and can re-initiate the last action or actions not executed by the card.

A description will now be given, in relation to Figure 3, still repeating the same example, of what happens when a cutoff occurs during the acknowledgement message.

This case can be envisaged only where an acknowledgement message is provided with the

application. However, the same problem can occur when the action counter is sent back at the time of a request for a new transaction, or the sending of an information message.

In this case, at the time of the new transaction request, the current value of the action counter of the card CA' = CA+n is sent back.

The server compares this value CA' with its last stored value, that is to say CA. As CA' = CA+n, the server knows that the last n actions have indeed been undertaken and stores the new value of the action counter, that is to say CA+n, in order to be synchronised with the card.